

Controlled experiments in Software Engineering: an introduction

Filippo Ricca

Unità CINI at DISI, Genova, Italy

Filippo.ricca@disi.unige.it



Controlled experiments ...



Marco Torchiano's daughter

- ...are like chocolate ...
- once you try it you become addicted!

Outline of the presentation

■ Introduction:

- ◆ What is a controlled experiment
- ◆ Why should we perform controlled experiments in SE
- ◆ What is their importance
- ◆ How to conduct a controlled experiment
- ◆ The experiment process
- ◆ Research questions
- ◆ Experimental Design:
 - Experiment (Goal, Quality focus, Perspective, Main factor)
 - Context (Subjects and Objects) and hypotheses
 - ...

■ Finally:

- ◆ **Filippo Ricca, Massimiliano Di Penta, Marco Torchiano, Paolo Tonella, Mariano Ceccato and Corrado Aron Visaggio. *Are Fit tables really talking? a series of experiments to understand whether Fit tables are useful during evolution tasks.* In *Proceedings of the 30th International Conference on Software Engineering (ICSE 2008)*, pages 361-370. IEEE Computer Society, 10-18 May 2008.**

What is a Controlled Experiment?

- It is an experiment done in a laboratory environment

High level of control

Book

Experimentation in Software Engineering: an Introduction
C. Wohlin et al.

- Manipulating one or more variables and control all the other variables

Is it clear?

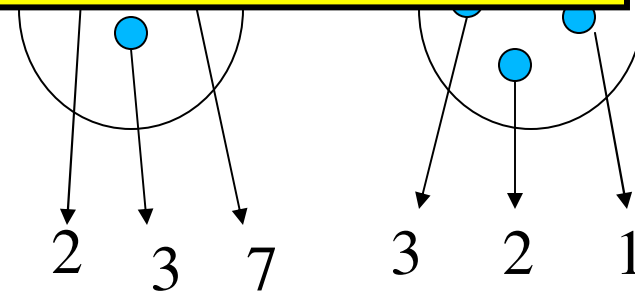
- The effect of manipulation is measured and statistical analysis applied.

Exemplifying ...

- We “lock in” a lab 6 Computer science students for 2 hours ...
- We want to understand if:
C++ is better than C.
- We randomly divide them in 2 groups
- We assign a simple task
- When students finished we run all the programs and count the “total number of defects”.
- We analyze the aggregated data and we compare the results.



We have conducted something similar to a controlled experiment!



$$\text{Mean C defects} = (2+3+7)/3=4$$
$$\text{Mean C++ defects} = (3+1+1)/3=2$$

Why should we perform an experiment in SE?

- Very general: to understand better Software engineering ...
- To get results regarding understanding, prediction, and improvement of software development/maintenance.
 - Is the technique A better than B?
- Empirical studies are important inputs to the decision-making in an organization.
 - Is the tool A better than B?

Experimental questions

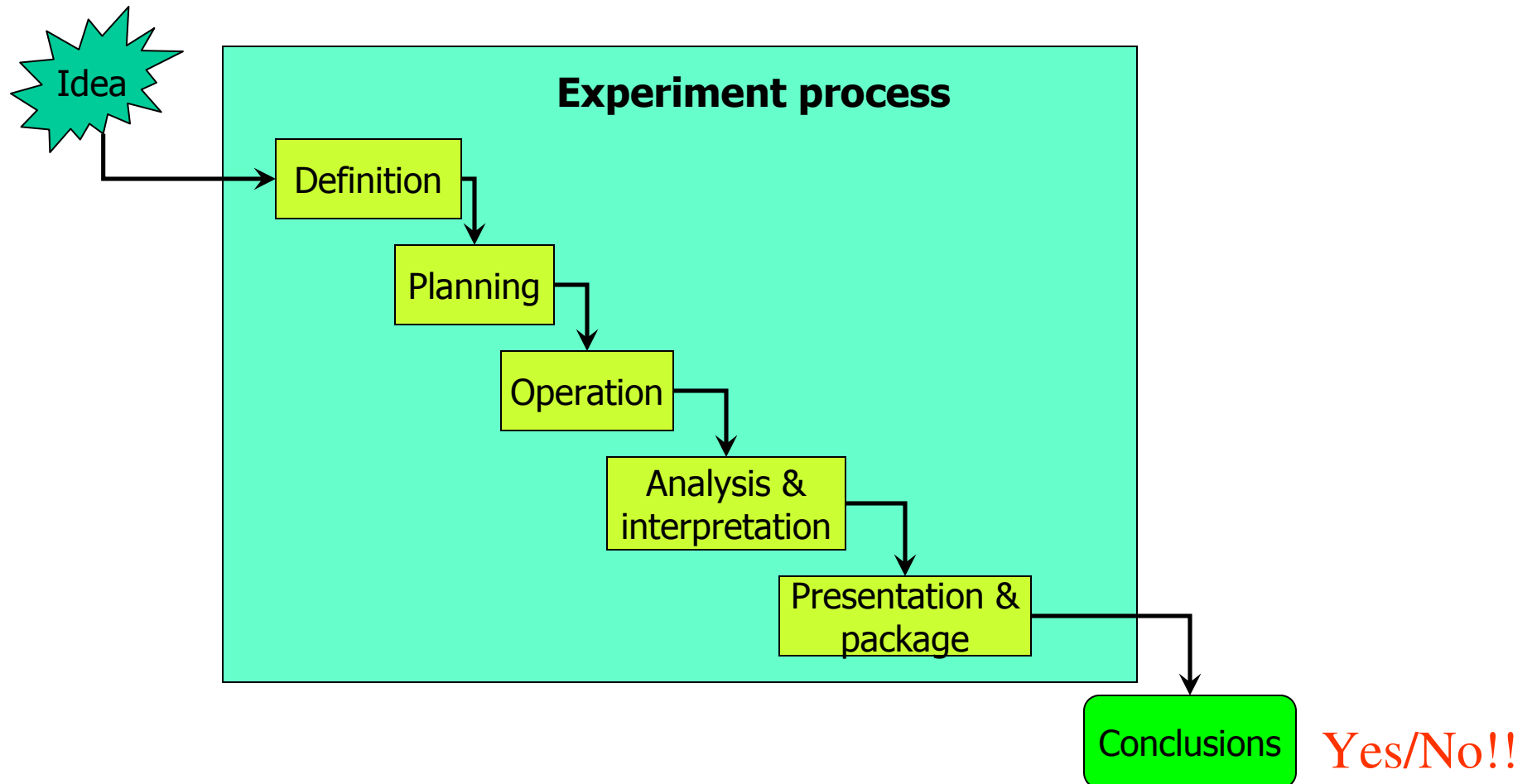
- **Experimental studies try to answer to experimental questions.**

Examples:

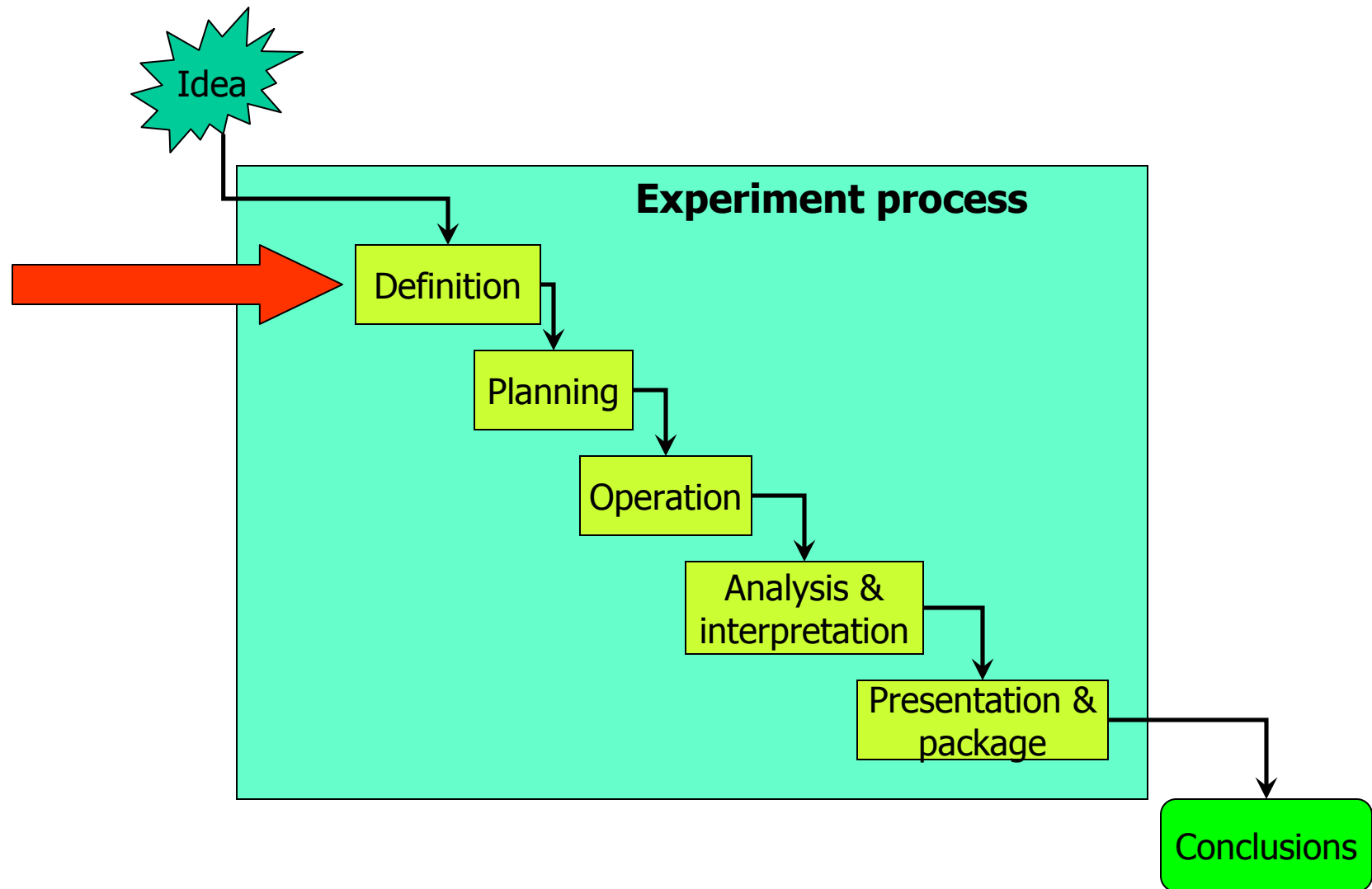
- | | |
|--|----------|
| ▪ Does 'stereotypes' improve the understandability of UML diagrams? | General |
| ▪ Does 'Design patterns' improve the maintainability of code? | |
| ▪ is 'Rational rose' more usable than 'Omondo' in the development of the software house X? | Specific |
| ▪ Does the use of 'JUnit' reduce the number of defects in the code of the industry Y? | |

Experiment process

“Is the technique A better than B ?”



Definition

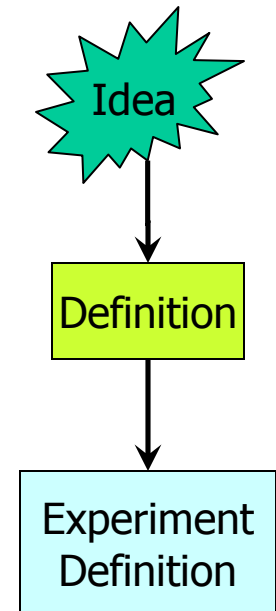


In this activity “what we want to test” has to be stated clearly.

Definition

Goal-definition-template

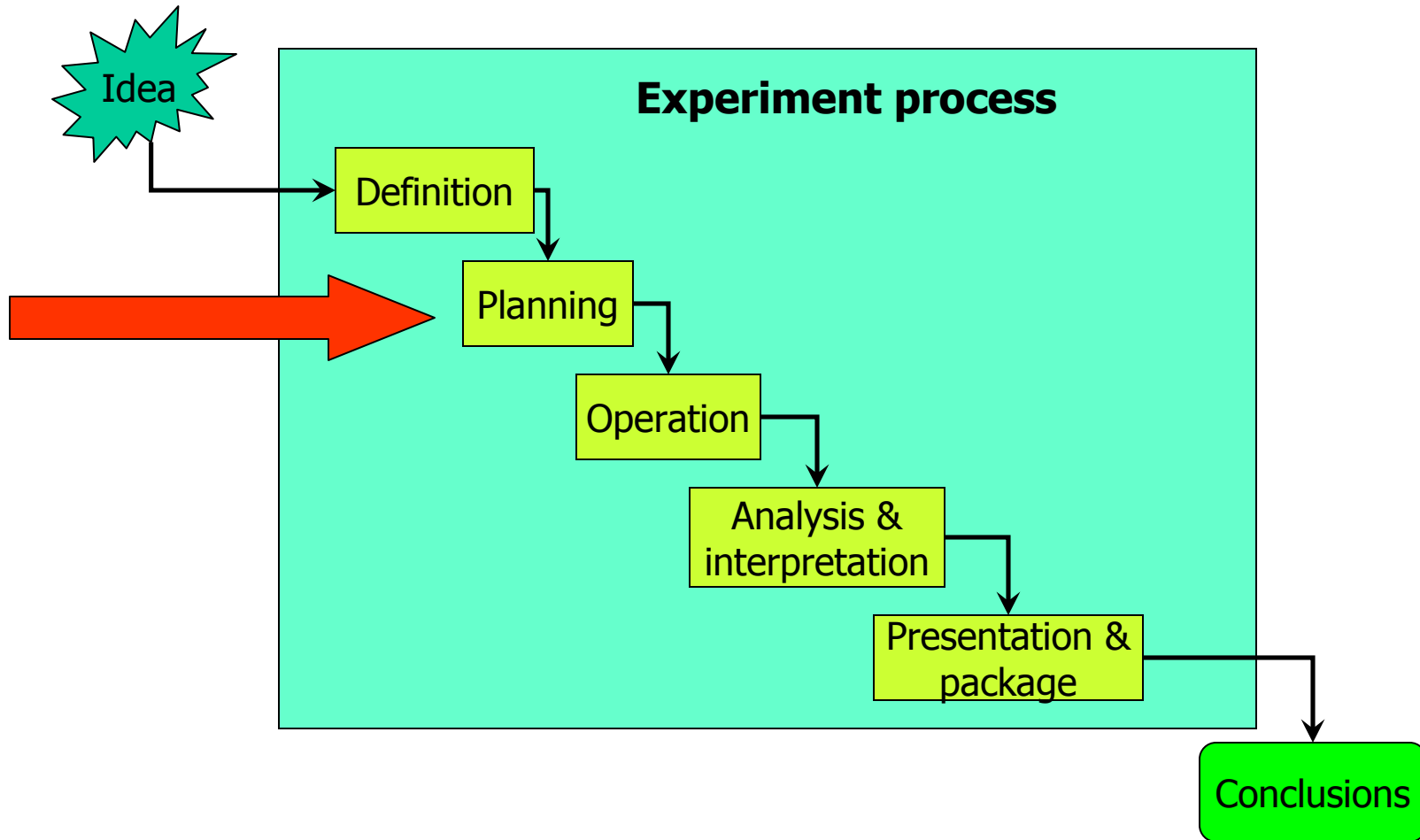
- **Object of the study:** is the entity that is studied in the experiment
 - ♦ e.g. code, process, design documents, ...
- **Purpose.** What is the intention of the experiment?
 - ♦ e.g. evaluate the impact of two different techniques
- **Quality focus.** The effect under study in the experiment.
 - ♦ e.g. cost, reliability, correctness...
- **Perspective.** Viewpoint from which the experiment results are interpreted.
 - ♦ e.g. developer, project manager, researcher, ...
- **Context.** The “environment” in which the experiment is run.
 - ♦ **subjects** and **objects**



C versus C++: definition

- **Object of the study.** Code (C and C++) of traditional applications
- **Purpose.** Evaluating if C++ is better than C (benefit of OO?)
 - ♦ i.e. if the number of defects in C++ programs is less than the number of defects in C code.
- **Quality focus (Which effect is studied?).** Correctness of the code
- **Perspective.** Multiple.
 - **Researcher:** evaluating which language is better;
 - **Project manager:** choosing between C and C++ (in his/her organization).
- **Context.**
 - **Subjects:** Computer science students.
 - **Object:** a simple application.
 - Find the first 'n' numbers in the Fibonacci sequence

Planning

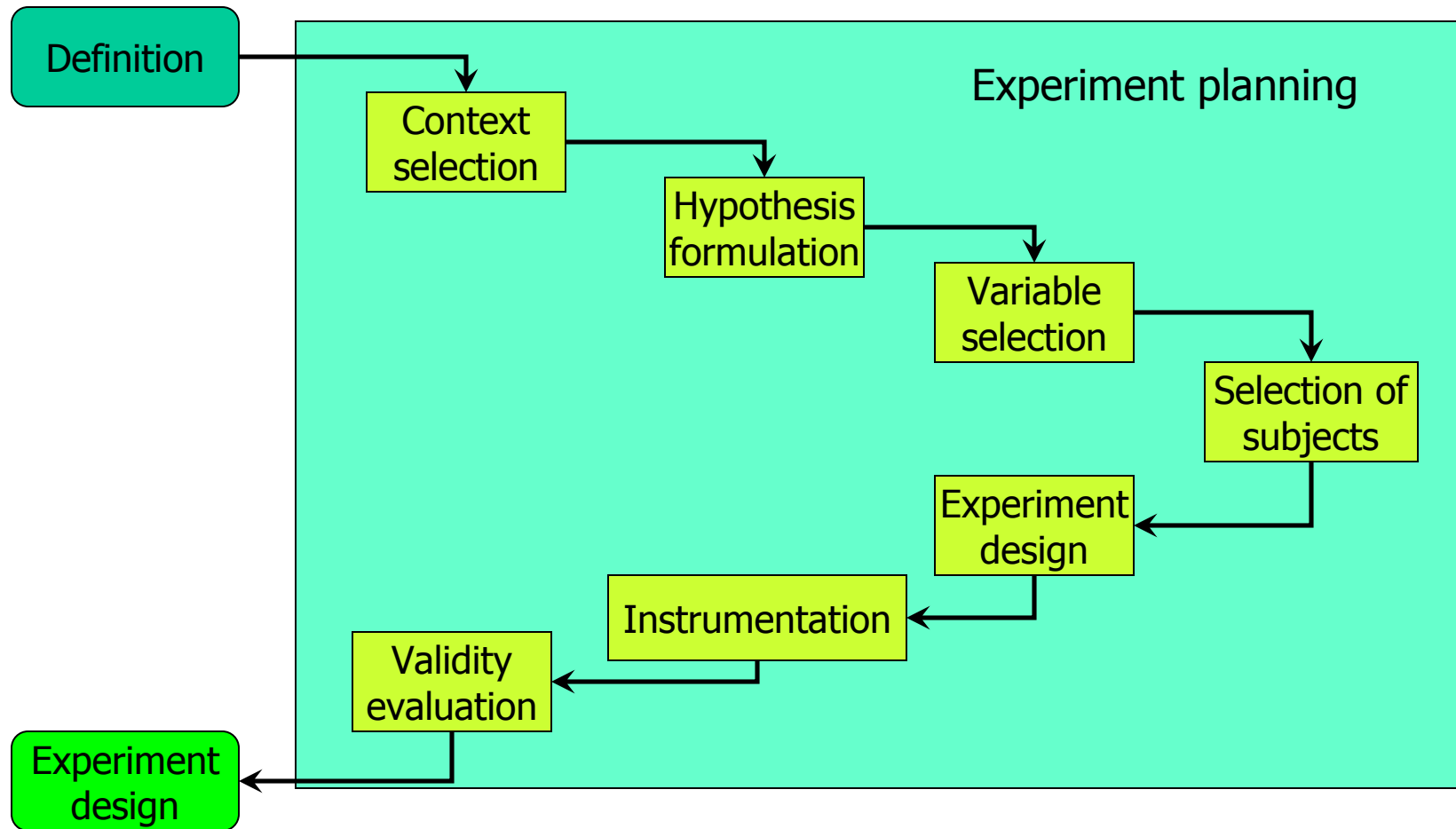


Planning

- **Definition** determines:
 - ♦ “Why the experiment is conducted”
- **Planning** prepares for
 - ♦ “How the experiment is conducted”.
- We have to state clearly:
 - ♦ Research questions
 - ♦ Context (subjects and objects)
 - ♦ Variables
 - ♦ Metrics
 - ♦ Design of the experiment
- The result of the experiment can be disturbed or even destroyed if not planned properly ...

Activity very important!

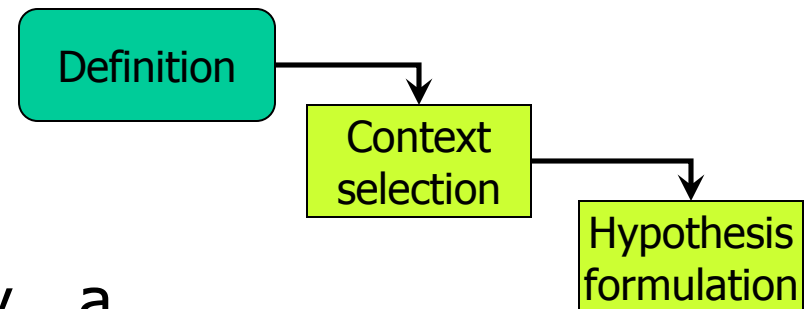
Planning phase "overview"



Planning activities (1)

- **Context selection**. We have four dimensions:

- ◆ off-line vs. on-line
- ◆ student vs. professional
- ◆ toys vs. real problems
- ◆ specific context (i.e. only a particular industry) vs. general context.

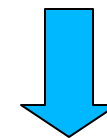


- **Hypothesis formulation**.

The hypothesis of the experiment is stated formally, including a

- ◆ null hypothesis
- ◆ alternative hypothesis.

Experimental questions

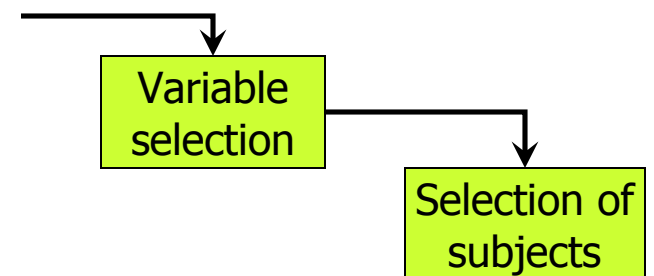
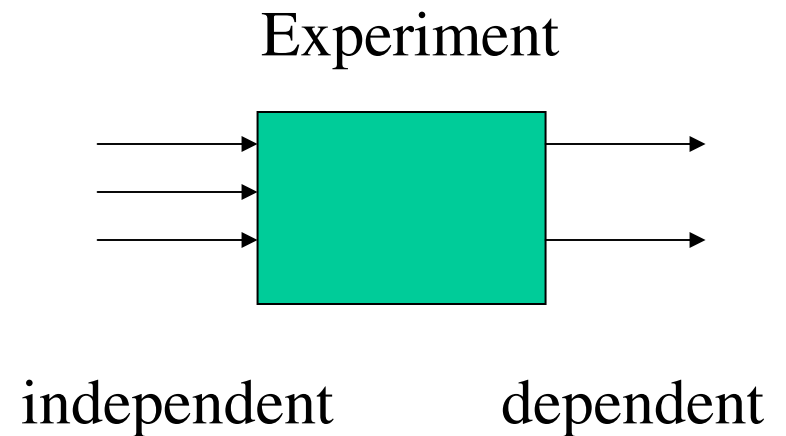


transformed

hypotheses

Planning activities (2)

- **Variables selection.** Determine independent variables (inputs) and dependent variables (outputs).
 - ♦ The effect of the treatments is measured by means of the dependent variables.
- **Selection of subjects.** In order to generalize the results to the desired population, the selection must be representative for that population.
 - ♦ The size of the sample also impacts the results when generalizing.



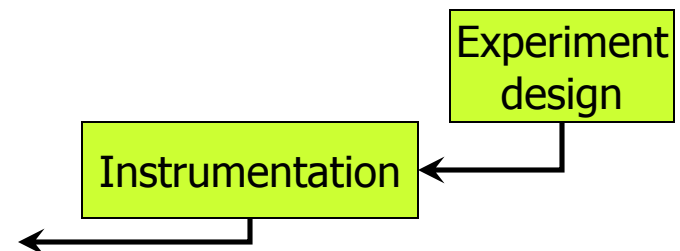
Students or professionals?

Planning activities (3)

- **Experiment design**. How to group subjects and how to apply treatments to them. Statistical analysis methods depend on the chosen design:
 - one factor with two treatments,
 - one factor with more than two treatments
 - ...
- **Instrumentation**. In this activity guidelines are decided to guide the participants in the experiment. Material is prepared and metrics decided:
 - Training
 - Questionnaires
 - Diagrams

“One factor with two treatments”

Subjects	C	C++
1	X	
2		X
3		X
4	X	
5		X
6	X	



Planning activities (4)

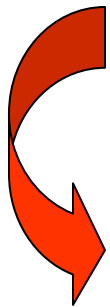
- **Validity evaluation.**

fundamental question:

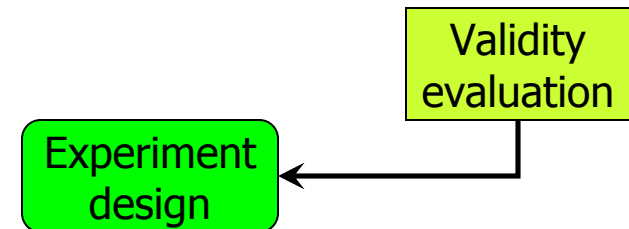
how valid the results are?

- ♦ **External validity:**

can the result of the study be generalized outside the scope of our study?



Threats to validity. Compiling a list of possible threats ...



Threats to validity

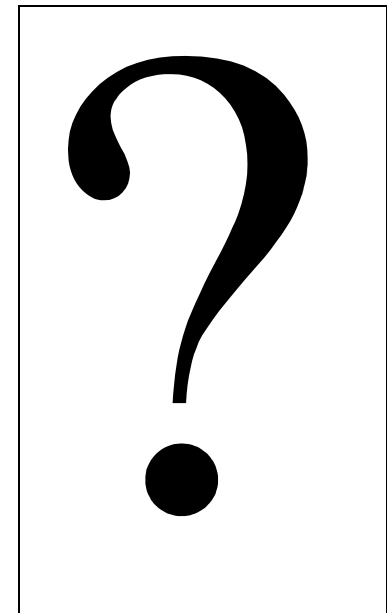
Examples:

- Experiment badly designated (materials, ...)
- Subjects not prepared to face the experiment (e.g., guidelines insufficient)
- Subjects anxiety (e.g., evaluation)
- Researchers may influence the results by looking for a specific outcome
- If the group is very heterogeneous there is a risk that the variation due to individual differences is larger than due to the treatment
 - ◆ Subjects background should be the same ...
- Confounding factors (e.g., variables not controlled)
- Small sample size → Low statistical power
- ...

Hypothesis

From Wikipedia:

- A **hypothesis** (from Greek *ὑπόθεσις*) consists either of a suggested explanation for a phenomenon or of a reasoned proposal suggesting a possible correlation between multiple phenomena.
- The term derives from the Greek, *hypotithenai* meaning "to put under" or "to suppose"
...



Hypotheses formulation

Two hypothesis have to be formulated:

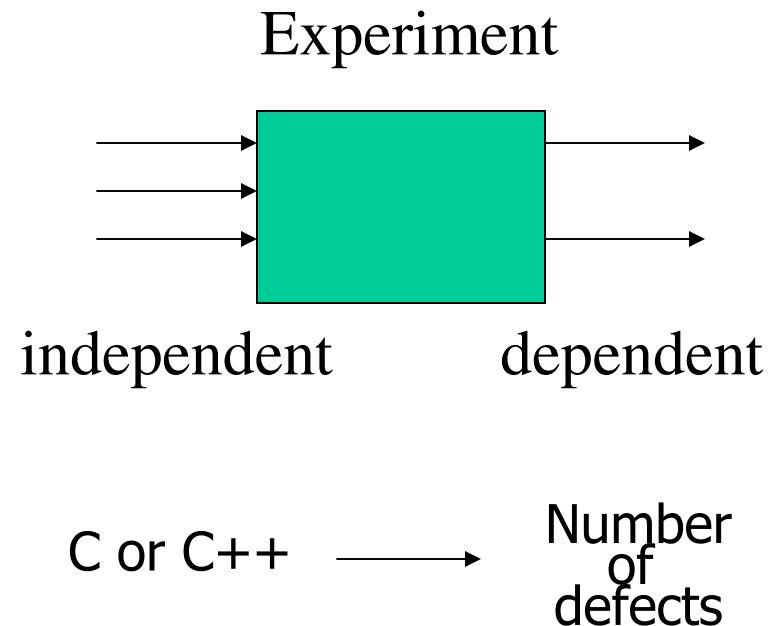
- 1. H_0 . The null hypothesis.** H_0 states that there are no real underlying trends in the experiment
 - ♦ the only reasons for differences in our observations are coincidental.
 - ♦ this is the hypothesis that the experimenter wants to reject with as high significance as possible.
- 2. H_1 . The alternative hypothesis.** This is the hypothesis in favor of which the null hypothesis is rejected.

H_0 : C++ programs contains on average the same number of defects as the C programs.

H_1 : C programs contains on average more defects than C++ programs.

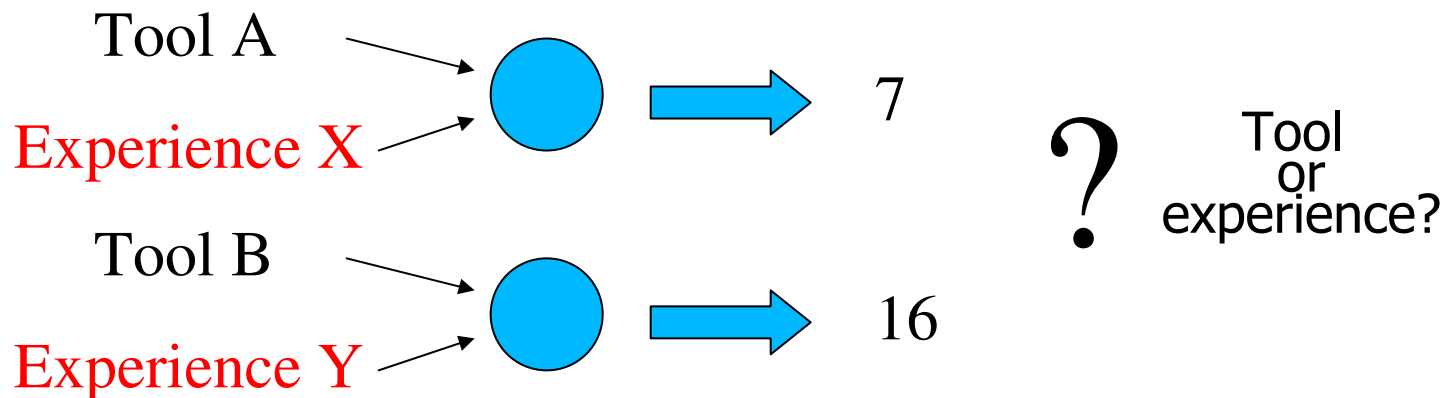
Variables selection

- The **independent variables** (inputs) are those variables that we can control and change in the experiment.
 - ♦ they describe the treatments
 - ♦ They describes the variables for which the effects should be evaluated.
- The **dependent variables** (outputs) are the response variables that describe the effects of the treatments described by the independent variables.
 - ♦ Often there is only one dependent variable and it should therefore be derived directly from the hypothesis



Confounding factors

- **Pay attention to the confounding factors!**
- A confounding factor:
 - ◆ is a variable that can hide a genuine association between factors
 - ◆ can confuse the conclusions of the experiment.
- For example, it may be difficult to tell if a better result depends on the tool or the experience of the user of the tool ...



C versus C++

- **Research question:** C++ is better than C?
- **Null hypothesis:** There is not difference in using C++ or C.
- The **independent variable** of interest in this study is the choice of programming language (C++ or C).
- **Dependent variables**
 - ◆ Total time required to develop programs
 - ◆ Total time required for testing
 - ◆ Total number of defects
 - ◆ ...
- Potential **confounding factor:** different experience of the programmers
 - ◆ e.g., they used C for 5 years and C++ only for 1 year

Pay attention!: Experiment not valid with confounding factors ...

Standard design types

- One factor with two treatments
- One factor with more than treatments
- Two factors with two treatments
- More than two factors each with two treatments
- ...

- ◆ The design and the statistical analysis are closely related.
- ◆ **We decide the design type considering:** objects and subjects we are able to use, hypothesis and measurement chosen.

One factor with two treatments (1)

- **Example:**
C++ is better than C?
- **We want to compare the two treatments against each other.**
- Factor = language chosen.
- Treatments = C/C++.

Completely randomized design

Subjects	C++	C
1	X	
2		X
3		X
4	X	
5		X
6	X	

If we have the same number of subjects per treatment the design is balanced

One factor with two treatments (2)

Completely randomized design

- ◆ **Example:**

C++ is better than C?

- ◆ The dependent variable is the number of defects found in the code.
- ◆ To understand if C++ is better than C we use:
 μ .

Subjects	C++	C
1	X	
2		X
3		X
4	X	
5		X
6	X	

μ_i = The average of defects for treatment i

Hypothesis: $H_0: \mu_1 = \mu_2$

$H_1: \mu_1 < \mu_2$ 27

We need more subjects ...

Two factors with two treatments

- The experiments gets more complex.

- **Example:**

C++ is better than C?

But considering also the development toolkit ...

	C	C++
Eclipse	Subjects: 4, 6	Subjects: 1, 7
NEdit	Subjects: 2, 3	Subjects: 5, 8

Factor A: language

Factor B: IDE

treatments: C, C++

treatments: Eclipse C/C++, NEdit

Metrics (dependent variables)

- How to measure the collected data?

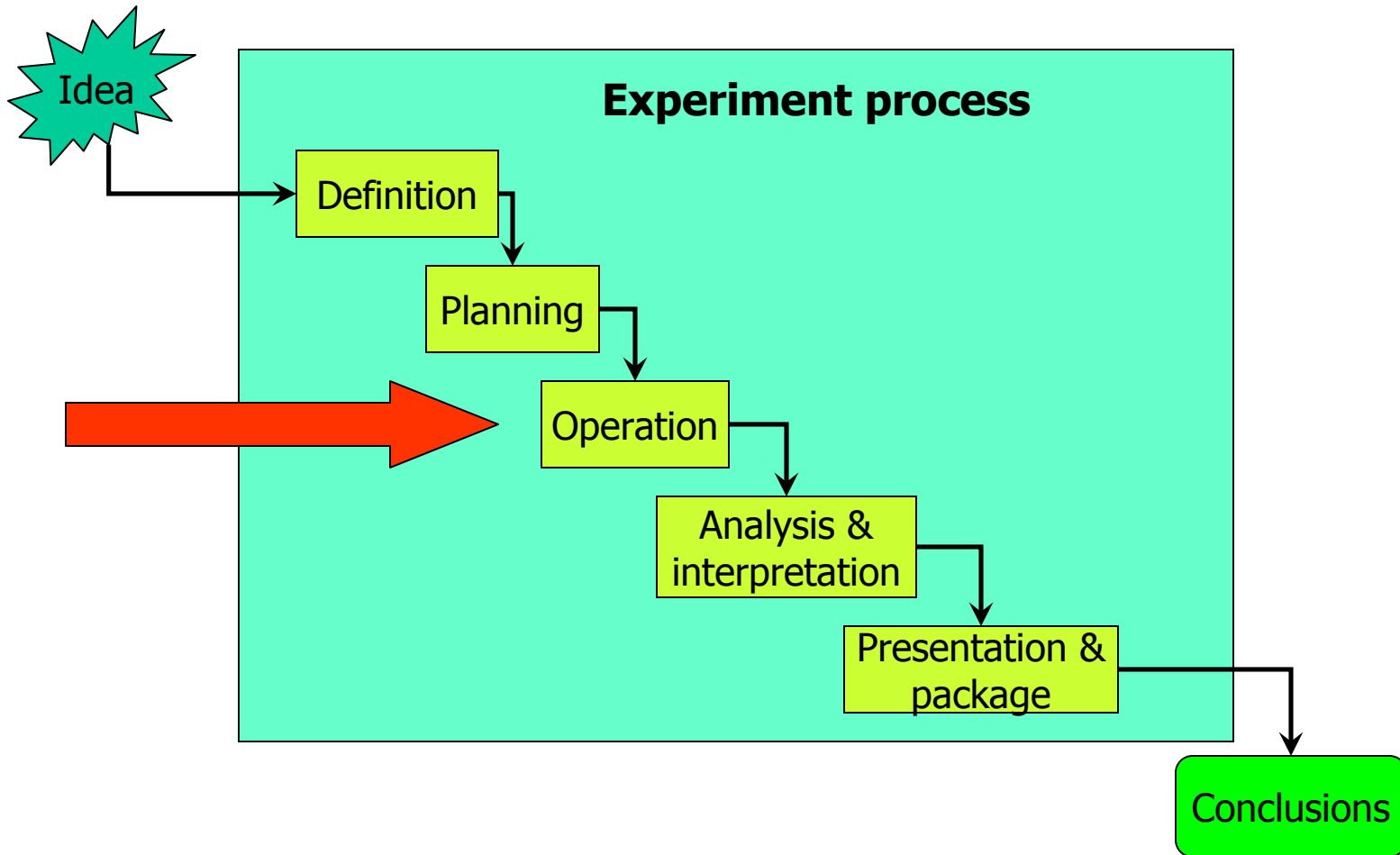
Using metrics ...

- Metrics reflect the data that are collected from experiments
- They are decided at “design time” of the experiment and computed after the entire experiment has ended.
- Usually, they are derived directly from the research questions (or hypotheses).

Metrics: examples

- **Question:** Does the design method A produce software with higher quality than the design method B?
 - ♦ **Metric:** number of faults found in the development.
- **Question:** Are OO design documents easier to understand than structured design documents?
 - ♦ **Metric:** percentage of questions that were answered correctly.
- **Question:** Are Computer science students more productive (as programmers) than Electronic engineers?
 - ♦ **Metric:** number of line of codes per total development time

Operation

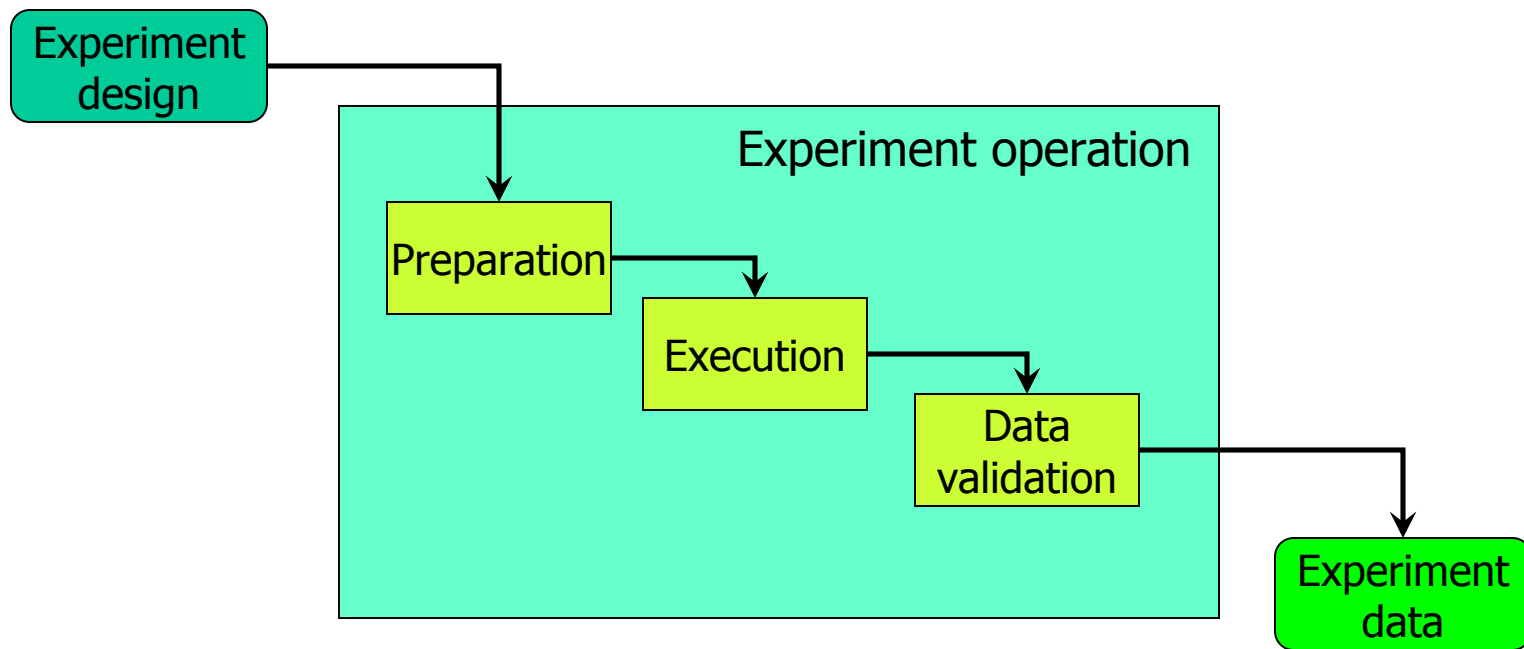


Operation

- After Definition and Planning: the experimenter actually meets the subjects.
- Treatments are applied to the subjects.
- Controlling subjects
- Data collection.
- Even if an experiment has been perfectly designed and the data are analyzed with the appropriate methods everything depends on the operation ...



Operation Steps

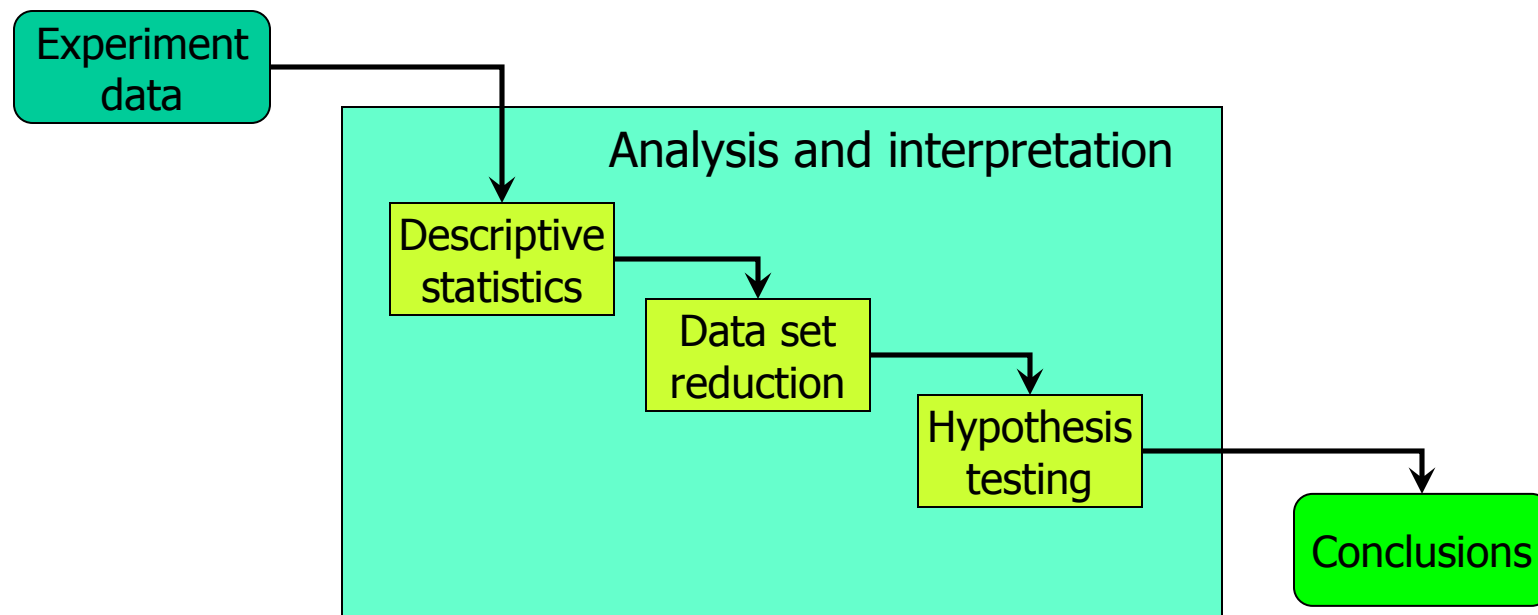


Preparation, execution, data validation

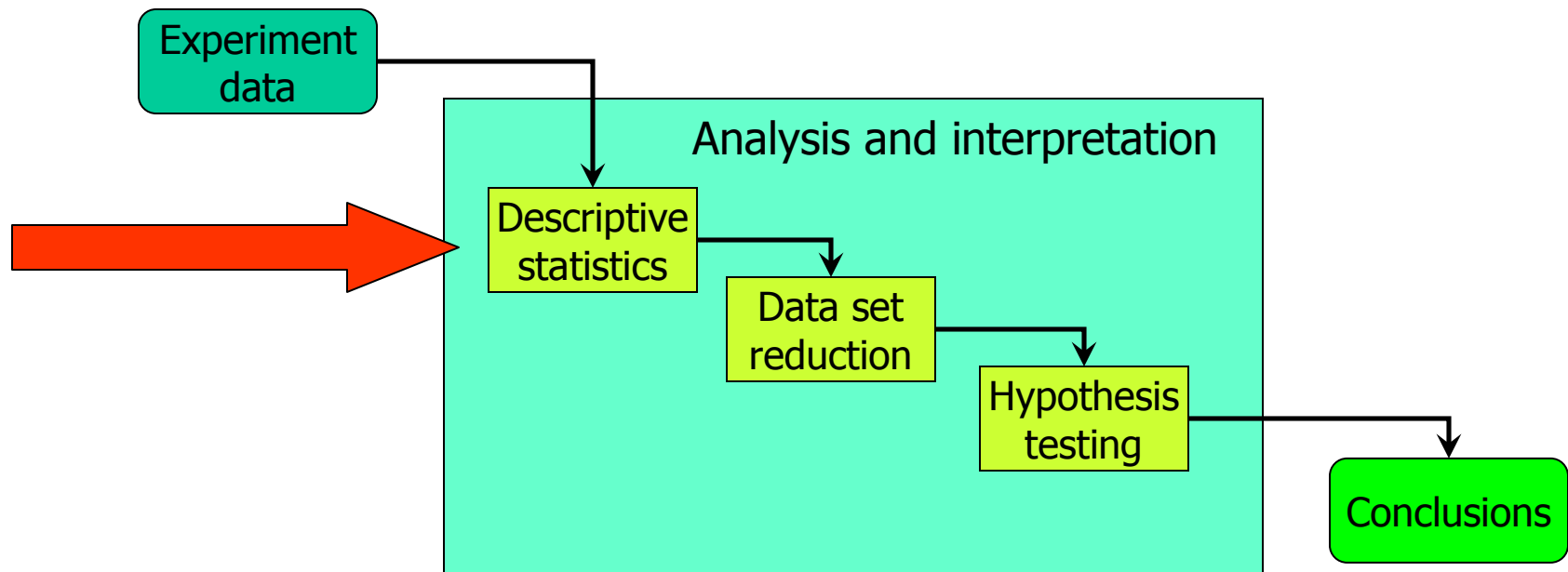
- **Preparation:** before the experiment can be executed, all instruments must be ready
 - ◆ forms, tools, software, ...
 - ◆ **Participants must be formed to execute the task!**
- **Execution:** Subjects perform their tasks according to different treatments and data is collected.
- **Data validation:** the experimenter must check that the data is reasonable and that it has been collected correctly.
 - ◆ *Have participants understood the task?*
 - ◆ *Have participants participated seriously in the experiment?*

Analysis and interpretation

- ◆ After collecting experimental data in the operation phase, we want draw conclusions based on this data.



Descriptive statistics

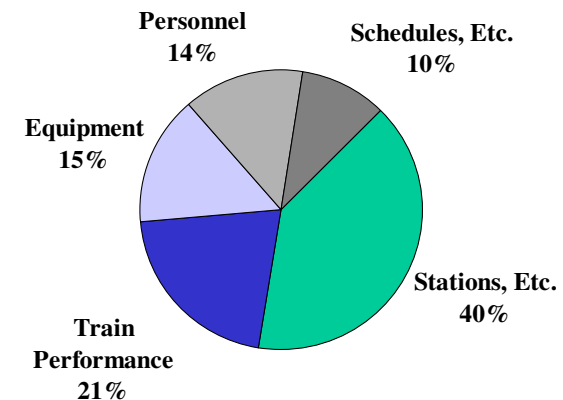
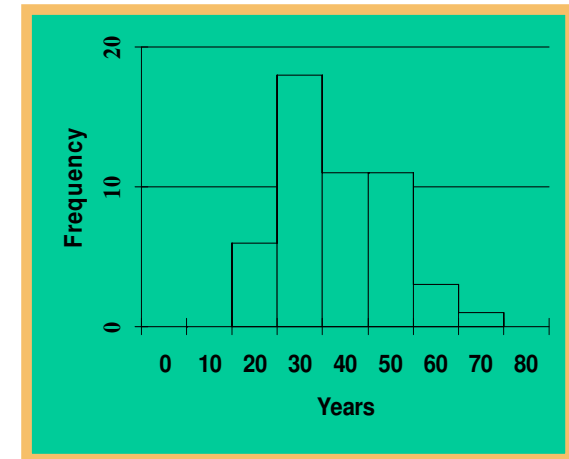
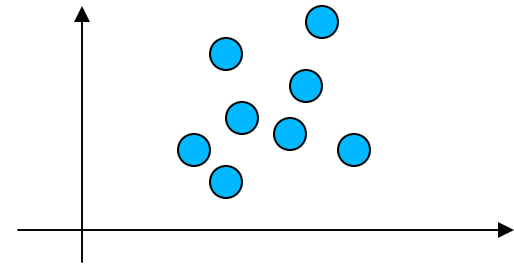


Descriptive statistics

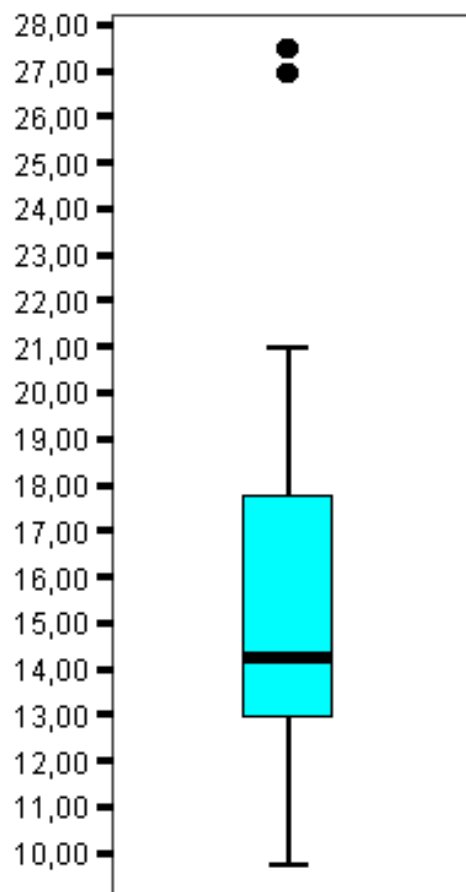
- Descriptive statistics deal with:
 - ◆ the presentation and numerical processing of a data set.
- Descriptive statistics may be used:
 - ◆ to describe and graphically present interesting aspects of the data set.
 - ◆ before carrying out hypothesis testing, in order to
 - better understand the nature of the data and
 - to identify abnormal data points (outliers).

Descriptive statistics

- Measures of **central tendency** (mean, median, mode, ...)
- Measures of **dispersion** (variance, standard deviation, ...)
- Measures of **dependency** between variables.
- **Graphical visualization** (scatter plots, histograms, pie charts, ...)



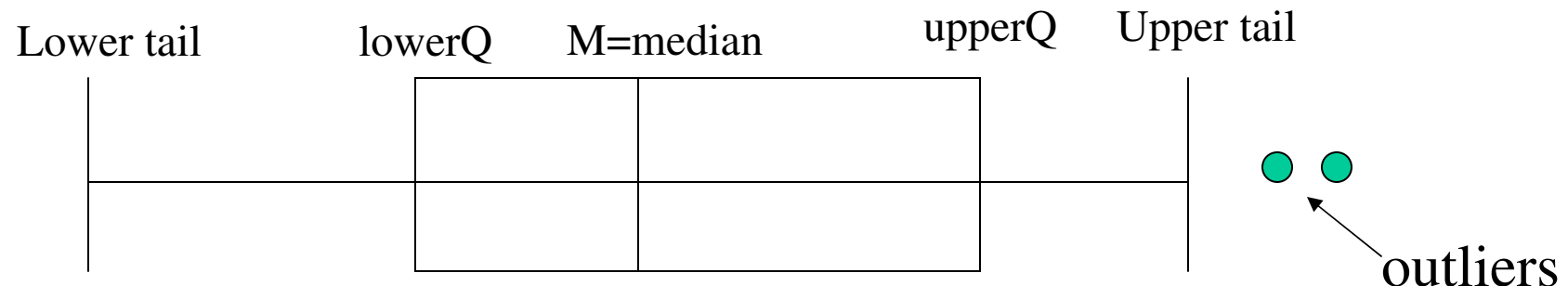
Box Plots



- ◆ Box plots are a graphical visualization technique used to visualize a data set.
- ◆ Box plots are good for visualizing the dispersion.
- ◆ **Example** (20 data):
27, 19, 12, 13, 21, 13, 19, 10, 12, 13, 16, 12, 14, 17, 13, 15, 14, 28, 14, 1.
 - ◆ Median is 14
 - ◆ 50% of the numbers are in the central rectangle (between 13 and 17)
 - ◆ 90% of the number are between the Whiskers (between 10 and 21)
 - ◆ 27 and 28 are the outliers

Box Plots: precise definition

- ◆ M = median
- ◆ lowerQ = the median of the values that are less than M
- ◆ upperQ = the median of the values that are greater than M
- ◆ Upper tail = $\text{upperQ} + 1.5(\text{upperQ} - \text{lowerQ})$
- ◆ Outliers = points external to [Lower tail, Upper tail]



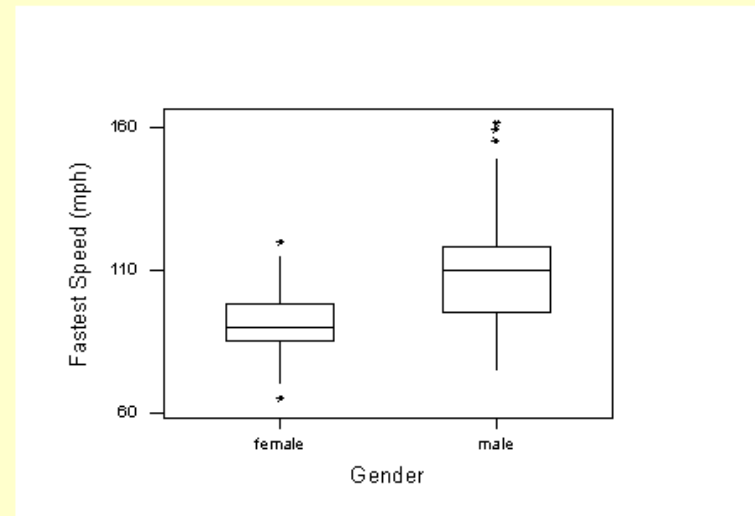
Box Plots

Guidelines for comparing boxplots

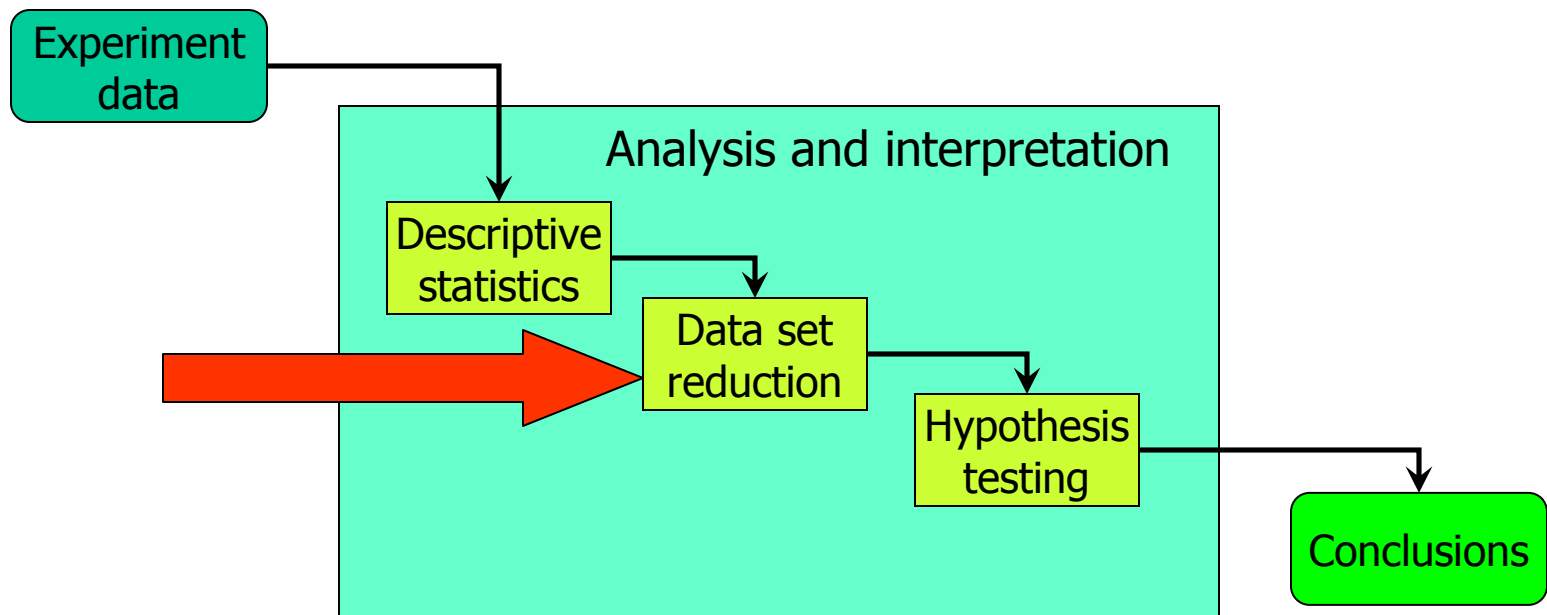
1. Compare the respective medians.
2. Compare the box lengths to compare dispersion.
3. Look for signs of skewness. Data are not symmetric if the median is not in the middle of the box
4. Look for potential outliers.

Box plots are used to compare samples

Using Box Plots to Compare

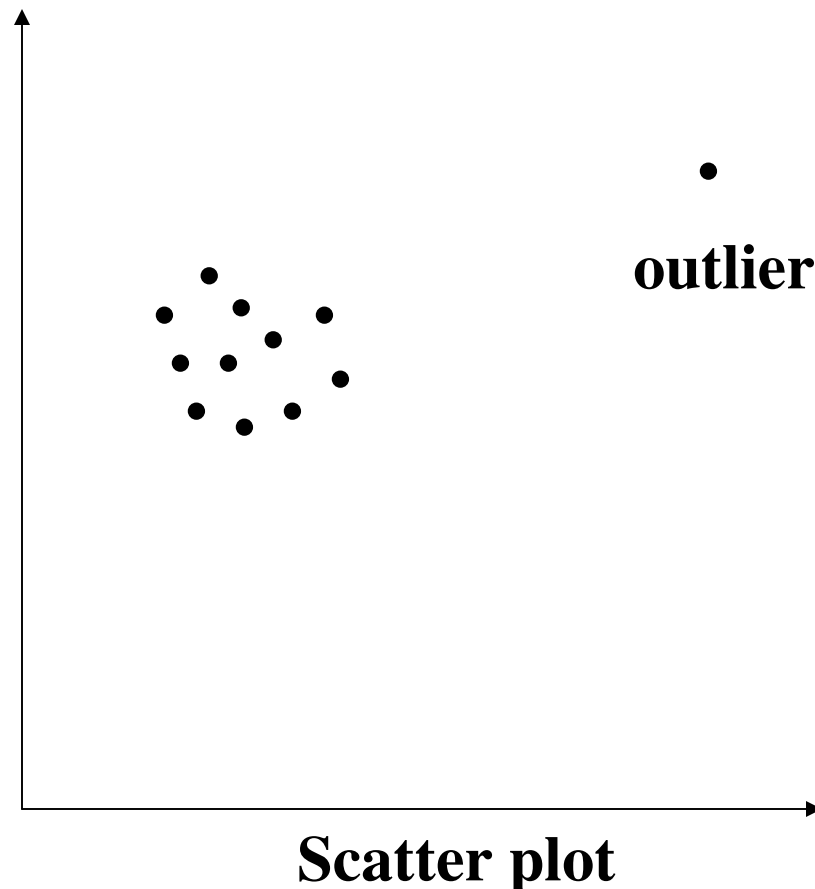


Data set reduction



Outlier analysis

- **Outlier** is a point that is much larger or much smaller than one could expect looking at the other points.
- A way to identify outliers is to draw **scatter plots**.
- There are statistical methods to identify outliers.



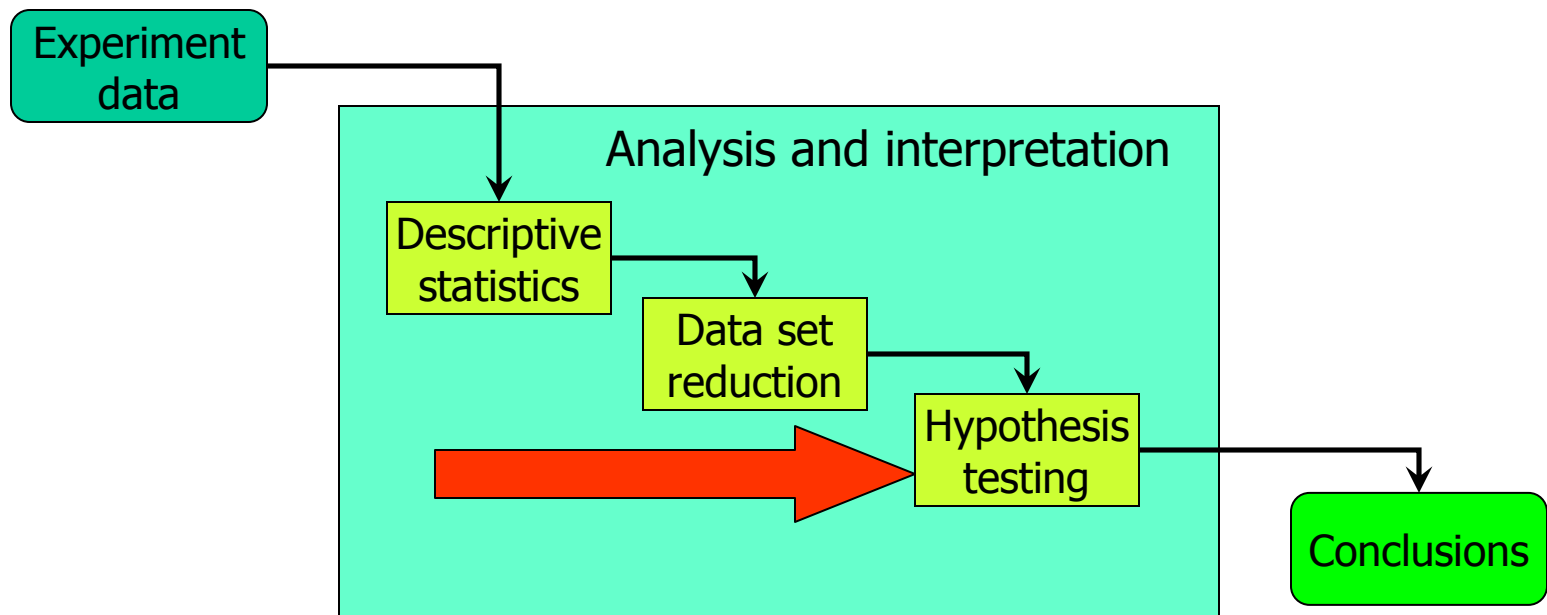
Data set reductions

- When outliers have been identified, it is important to decide

what to do with them?

- If the outlier is due to a strange or rare event that never will happen again, the point can be excluded.
 - ♦ Example. The task was not understood.
- If the outlier is due to a rare event that may occur again, it is not advisable to exclude the point. There is relevant information in the outlier.
 - ♦ Example: a module that is implemented by inexperienced programmers.

Hypothesis testing



Descriptive vs. Inferential Statistics

- **Descriptive Statistics** — using data gathered on a group to describe or reach conclusions about that same group only.
- **Inferential Statistics** — using sample data to reach conclusions about the population from which the sample was taken.

Making Inference

- Research Question

C++ is better than C
(the number of defects in C++ is
less than in C)

- Choose a sample

I take four
programs ...

- Run the experiment

I observe the number of
defects running C++
and C programs

- Collect Results

A: 3 5 4 4 *mean=4*
B: 8 4 6 6 *mean=6*

Statistical test (1)

Are obtained results statistically significant?

Can we conclude that C++ is better than C?
(or the observed difference is due by chance ...)

In general ...

- Properties of a **sample** can be extended to the **entire population**?
- **Statistical test** are used to answer to this type of questions.

Statistical test (2)

- t-test
- Chi-2
- Anova
- Mann Whitney
- Wilcoxon
- ...

- There are a number of different **statistical tests** described in the literature that can be used to evaluate the outcome of an experiment.
- They are all based on Statistical Hypotheses.

How to reason? The scientific method

- We make a hypothesis solely for the purpose of having it rejected.
- If we want to decide whether one alternative is better than another is, we formulate the hypothesis that *there is no difference between the two alternatives.*
- Any hypothesis that differs from the null hypothesis will be called an alternative hypothesis



Null
Hypothesis

Reductio ad absurdum (Latin for “reduction to the absurd”)

Example

Research hypothesis:

C++ is better than C?
(the number of defects in C++ is less than in C).

Null Hypothesis

$$H_0 : \mu_{c++} = \mu_c$$

The number of defects in C and C++ is the same

Alternative Hypothesis

$$H_a : \mu_{c++} < \mu_c$$

Pay attention: μ is the mean of the population!
(considering all possible programs ...)

Critical area

- A **statistical test** can be formulated in this way:

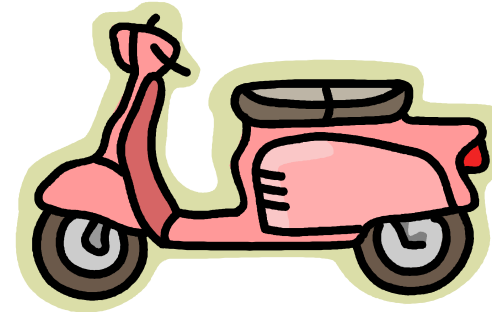
<p>If $t \in C$ reject H_0</p> <p>If $t \notin C$ do not reject H_0</p>

where **C** is the **critical area** and **t** is the value resulted from the experiment.

- A critical area **C** can have different shapes but usually it is an interval.
- If **C** consists of one interval (for example $K < a$) it is **one-sided**. If **C** consists of two intervals ($K < a$ and $K > b$ with $a < b$) it is **two-sided**.

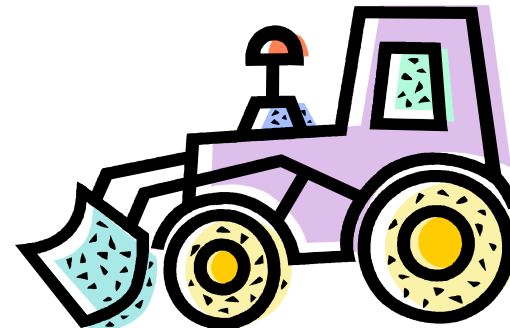
Example

- An experimenter observes a vehicle and wants to show that the vehicle is not a car.
- The experimenter knows that all cars have four wheels, but also that there are other vehicle than cars that have four wheels.
- **H₀: "the observed vehicle is a car"**
- **t: number of wheels**
- **Test: $t \leq 3$ or $t \geq 5$ reject H₀
 $t=4$ do not reject H₀**
- If it is observed that t is 4, it means that the H₀ cannot be rejected and no conclusion can be drawn. This is because there may be other vehicles than cars with four wheels.



H₀ is rejected

Critical area $C = \{1, 2, 3, 5, 6, 7, \dots\}$



H₀ can not be rejected!

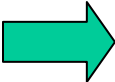
Hypothesis testing

- Is it possible to reject a certain null hypothesis?
- To answer at this question we have to use statistical tests (t-test, ANOVA, Chi-2, ...).
- The application of statistical tests to sample data gives the **p-value**.

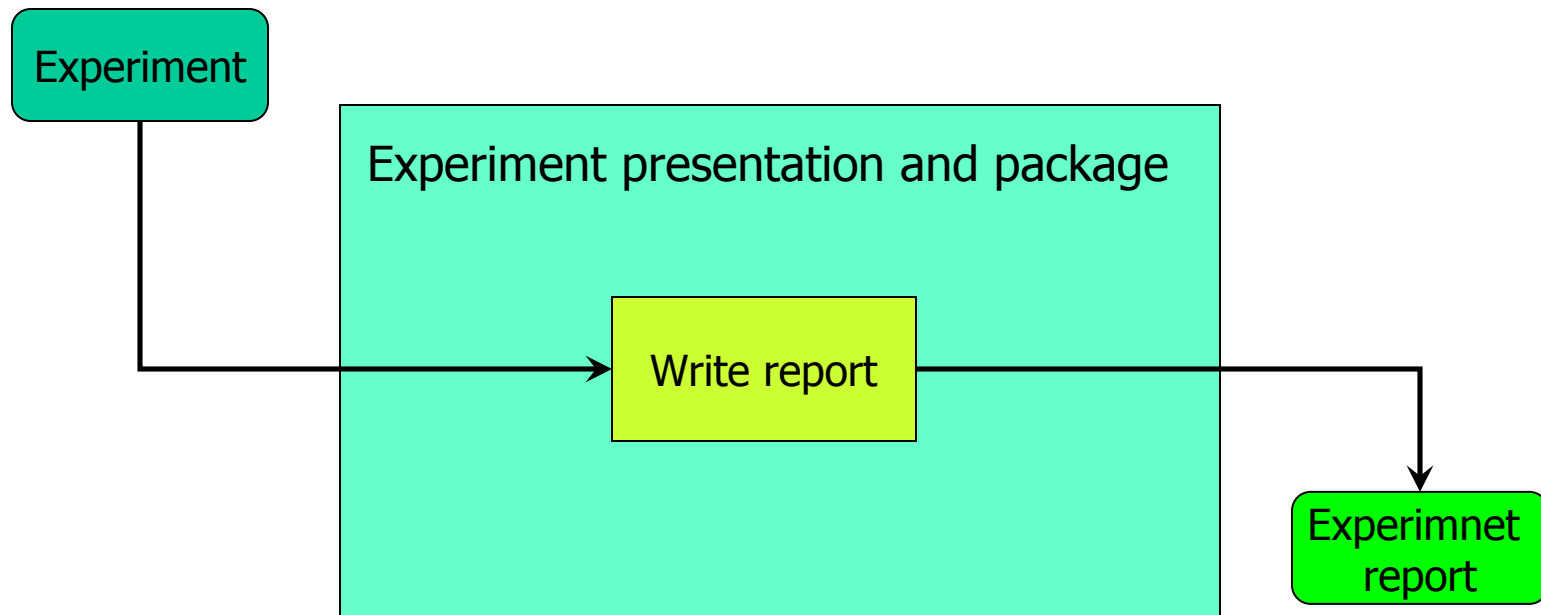
But what is the
p-value?

It a number (real) but we need
to understand its meaning!

P-value: Just an intuition ...

- The p-value indicates:
“**whether the result of the experiment is in the critical area**”
- If the **p-value** $<$ given threshold (α) then we can reject the null hypothesis.
- Usually $\alpha=0.05$ (95%)
- **Example:**
 $\text{p-value}=0.03 < \alpha$  **it is possible to reject H₀**
- **Statistically speaking**: 0.03 (3%) is the probability to commit an error, that is, the probability to reject H₀ when H₀ is true.
- We have the confidence of 97% of having made the right choice ...

Presentation and package



Are Fit Tables Really Talking?

A Series of Experiments to Understand whether Fit Tables are Useful during Evolution Tasks

Filippo Ricca¹, Massimiliano Di Penta²,
Marco Torchiano³, Paolo Tonella⁴, Mariano Ceccato⁴,

Corrado Aaron Visaggio²

¹ *Unità CINI at DISI, Genova, Italy*

² *RCOST - University of Sannio, Benevento, Italy*

³ *Politecnico di Torino, Italy*

⁴ *Fondazione Bruno Kessler-IRST, Trento, Italy*

Motivations

- **Usage of natural language to specify (change) requirements:**
 - ◆ current state of the practice
 - ◆ highly inaccurate
 - ambiguous, incomplete, inconsistent, silent, unusable, over-specific or verbose textual requirements [Meyer 1985].
 - ◆ error prone
 - 85% of the defects are estimated to originate from inadequate requirements [Young 2001].

- **“Agile movement” advocates [Melnik et al. 2004]:**
 - ◆ Acceptance test cases constitute an expressive form of documentation
 - ◆ Acceptance test cases are “talking” representation of the requirements

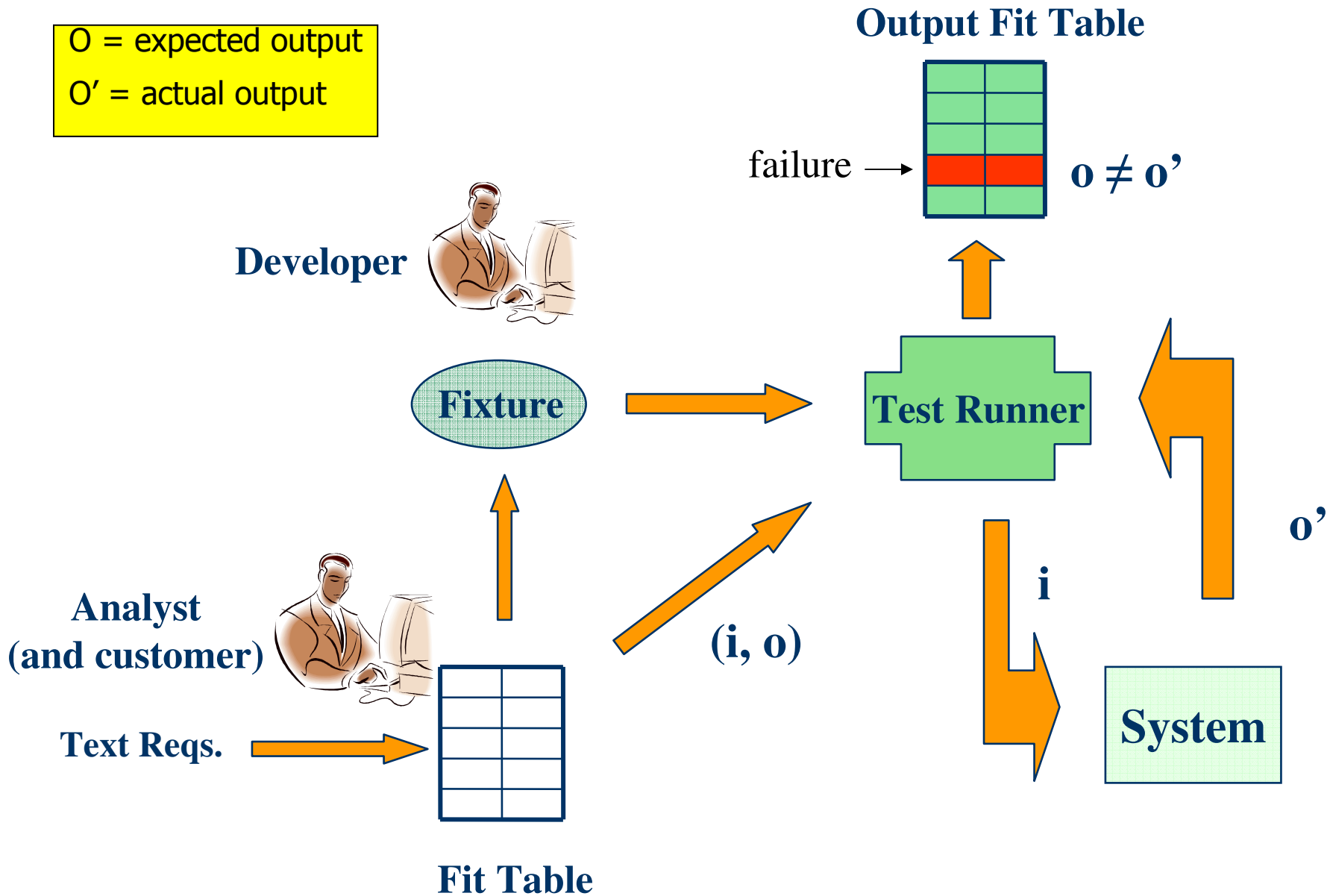
- **We focus on evaluating whether Fit acceptance tests**
 1. are **useful** in maintenance/evolution tasks
 2. **affect the time** needed for maintenance/evolution tasks

Framework for Integrated Test

- The most well-known open source **implementation of the table-based acceptance testing approach.**
 - ♦ A table represents a set of test cases.
- **Fit** lets:
 - ♦ customers/analysts write “executable” acceptance tests using simple HTML tables (**Fit tables**).
 - ♦ developers write “**Fixtures**” to link the test cases with the actual system itself.
 - ♦ to compare (**Test Runner**) these test cases with actual values, returned by the system, and highlights the results with colors.
 - **Red** for failures
 - **Green** for passed tests.

"The complete picture"

O = expected output
O' = actual output



"Core" Fit Tables

- Core Fit tables:
 - **Column Fit tables** for testing calculations
 - **Action Fit tables** for testing user interfaces or workflows;
 - **Row Fit tables** to check the result of a query
- (some) other Fit tables:
 - **Summary Fit tables**
 - **Html Fit tables**
 - **Command line Fit tables**
 - ...
- **We focused on core Fit tables**

Column Fit table

small bag price = 0.62

fit_tests.DiscountStructure			
small bags	beverage	discount	total price()
2	Coffee	0	1.24
4	Tea	0	2.48
5	Coffee	1	2.1
5	Tea	0	3.1
7	Coffee	1	3.34
7	Tea	0	4.34

Action Fit table

1 box = 60 small bags

fit.ActionFixture		
start	fit_tests.VerifySupply	
enter	type of product	Coffee
check	number of small bug remained	10
enter	number of box	5
press	buying boxes	
check	number of small bug remained	310

Text only vs. Text + Fit

LaTazza application

C
O
T
ch
be
is

RQ1:
Does the presence of Fit tables help programmers to improve the correctness of maintained code?

LaTazza application

Requirement: Change price
boxes
Vendor of boxes of beverages
changed his selling policy. Each five
boxes (of the same type) one
is given as a gift.

fit.ActionFixture

start | fit_tests.VerifySupply

RQ2:
Does the presence of Fit tables affect the time needed for a maintenance task?

check | cash account

504

"-" Text only group

"+" FIT group

Experiment

- **Goal:** Analyze the use of Fit tables with the purpose of evaluating their usefulness during maintenance tasks for different categories of users.
- **Quality focus (Which effect is studied?):** correctness of maintained code and maintenance time.
- **Perspective:** researchers, project managers
- **Main factor**
 - ◆ Availability (or not) of Fit tables:
 - **Treatments:** Text only vs. Text + Fit tables

Context and hypotheses

◆ Context:

- ◆ **Subjects:** 40 students from 3 courses
 - Exp I - Trento (14 Master students)
 - Exp II - Trento (8 PhD students)
 - Exp II - Benevento (18 Bachelor students working in pairs)
- ◆ **Objects (2 Java applications):**
 - **LaTazza** (application for a hot drinks vending machine)
 - 18 classes, 9 Reqs, 18 Fit tables, 1121 LOC
 - **AveCalc** (manages an electronic record book for master students)
 - 8 classes, 10 Reqs, 19 Fit tables, 1827 LOC

◆ Null Hypotheses:

- ◆ H_{0c} : the availability of Fit tables does not significantly **improve the correctness** of the maintained source code.
 - One-tailed
- ◆ H_{0t} : The availability of Fit tables does not significantly **affect the time** needed for the maintenance task.
 - Two-tailed

Design

- **Balanced experiment design**
 - two labs, 2 hours each.
- Subjects split into four groups
- Asked to complete **four maintenance tasks** with or without Fit tables.
 - “+” (**FIT group**) requirements and change requirements enhanced with Fit tables and fixtures
 - “-” (**Text only group**) textual requirements/change requirements only

	Group A	Group B	Group C	Group D
Lab 1	LaTazza +	LaTazza -	AveCalc -	AveCalc +
Lab 2	AveCalc -	AveCalc +	LaTazza +	LaTazza -

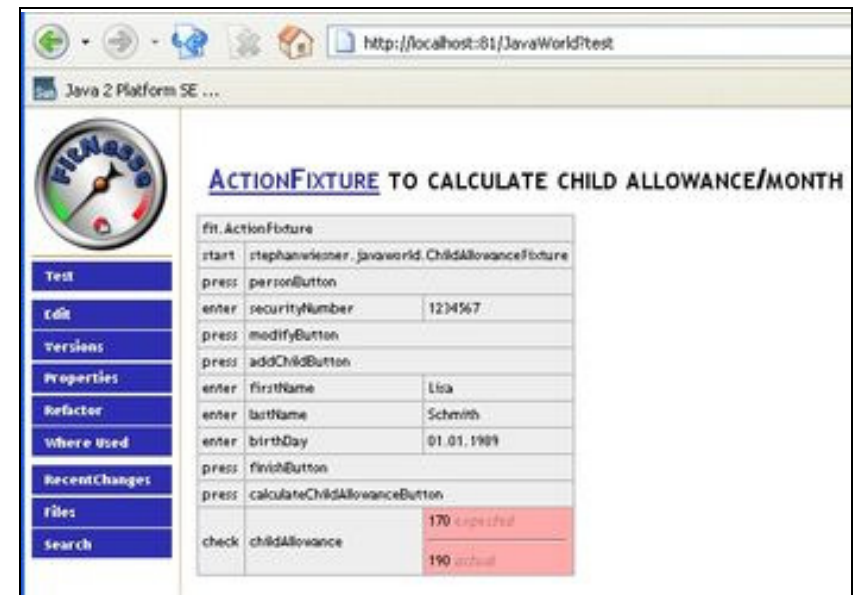
Material

Working setting:

- **Eclipse IDE**
- **Fitnessse plug-in** used to:
 - browse requirements
 - browse change requirements
 - execute Fit Test-cases (+)

Each subject received:

1. A short application description
2. Lab instructions
3. an Eclipse project containing:
 - the source code
 - the **Fitnessse Wiki** with reqs, change reqs and Fit tables (+)
4. a time sheet
5. a post experiment questionnaire



Procedure

After reading the documentation, for each change requirement:

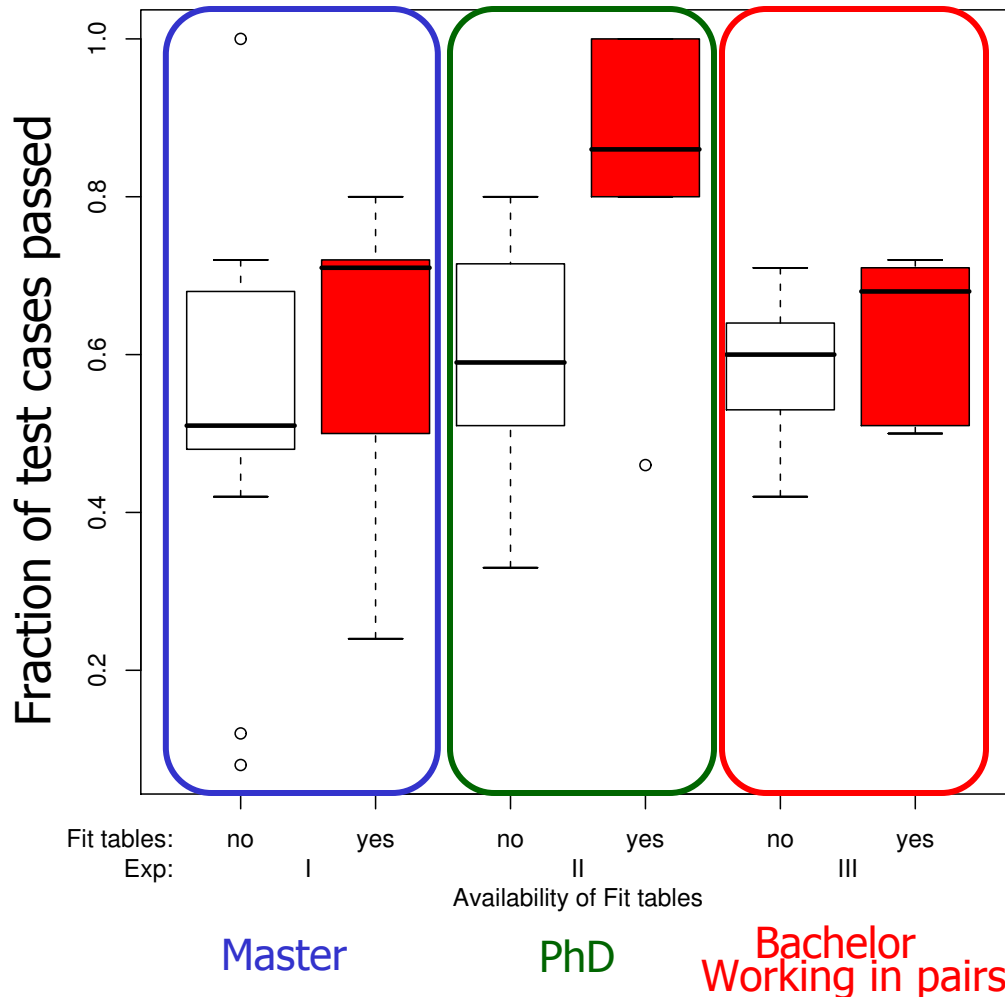
1. record the starting time
2. read the change requirement and look at Fit tables (+)
3. implement the change requirement
4. if Fit tables are available, run test cases
 - of the application requirements (for regression testing purposes)
 - of the change requirements
5. iterate steps 3 and 4 if necessary
6. record the completion time

**Subjects monitored to check
that the procedure were correctly followed**

Variables

- **Dependent Variables:**
 - ◆ **Code correctness**
 - assessed by executing a **JUnit test suite**
 - developed independently from Fit tables
 - composed by 25 (AveCalc) and 24 (LaTazza) test cases
 - devised using of the **category partitioning black-box strategy**
 - quantified as **percentage of test cases passed.**
 - ◆ **Time required to perform the maintenance task**
 - measured in minutes by relying on time sheets
- **Independent Variables:**
 - ◆ Main factor treatments: Text only Reqs vs Reqs + Fit tables

Results: Effect of main factor

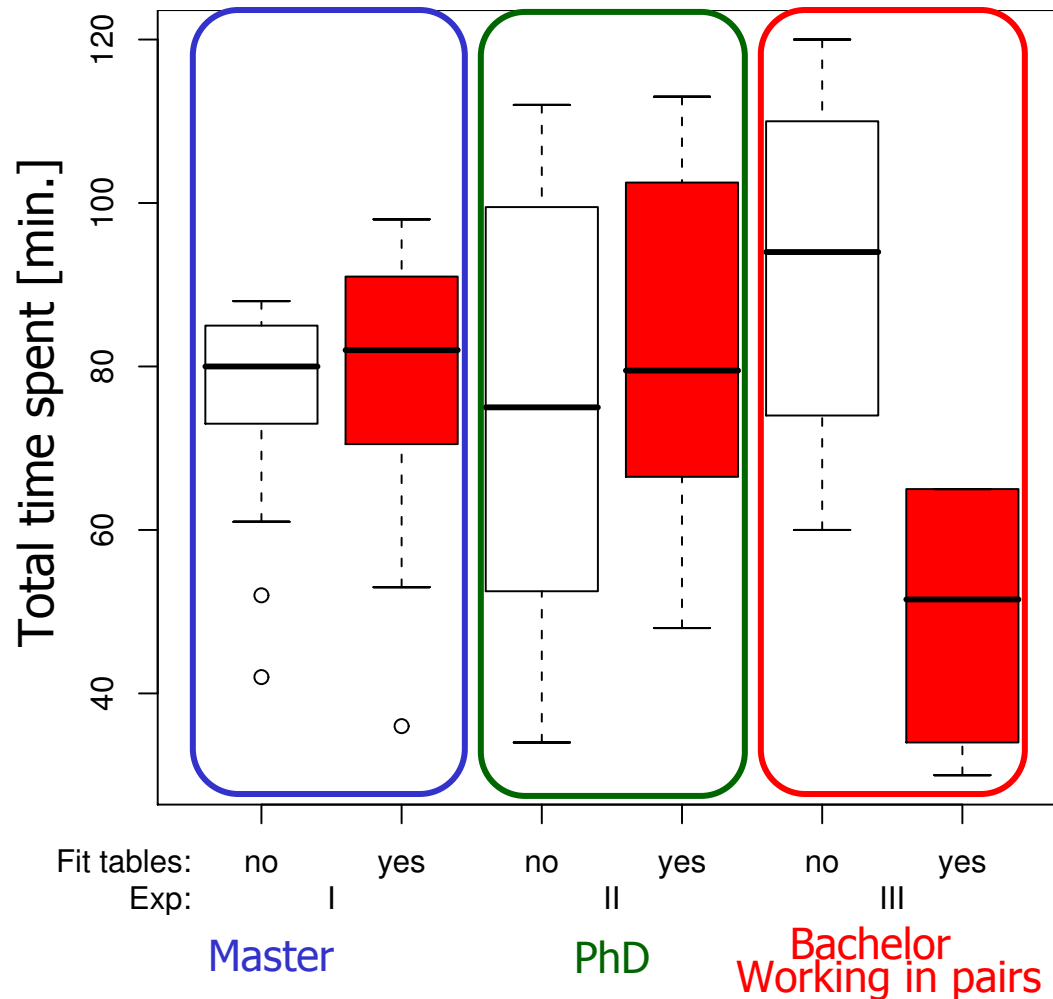


Red = having Fit tables

- ◆ Mann Whitney Test
Significant difference in
 - ◆ Exp I (p-value=0.04)
 - ◆ Exp II (p-value<0.01)
 - ◆ Overall (p-value<0.01)
- ◆ Wilcoxon test
 - ◆ Significant difference in Exp II (p-value=0.01).
 - ◆ Significant differences on the whole set (p-value=0.02)

Overall, we can reject H_{0c}

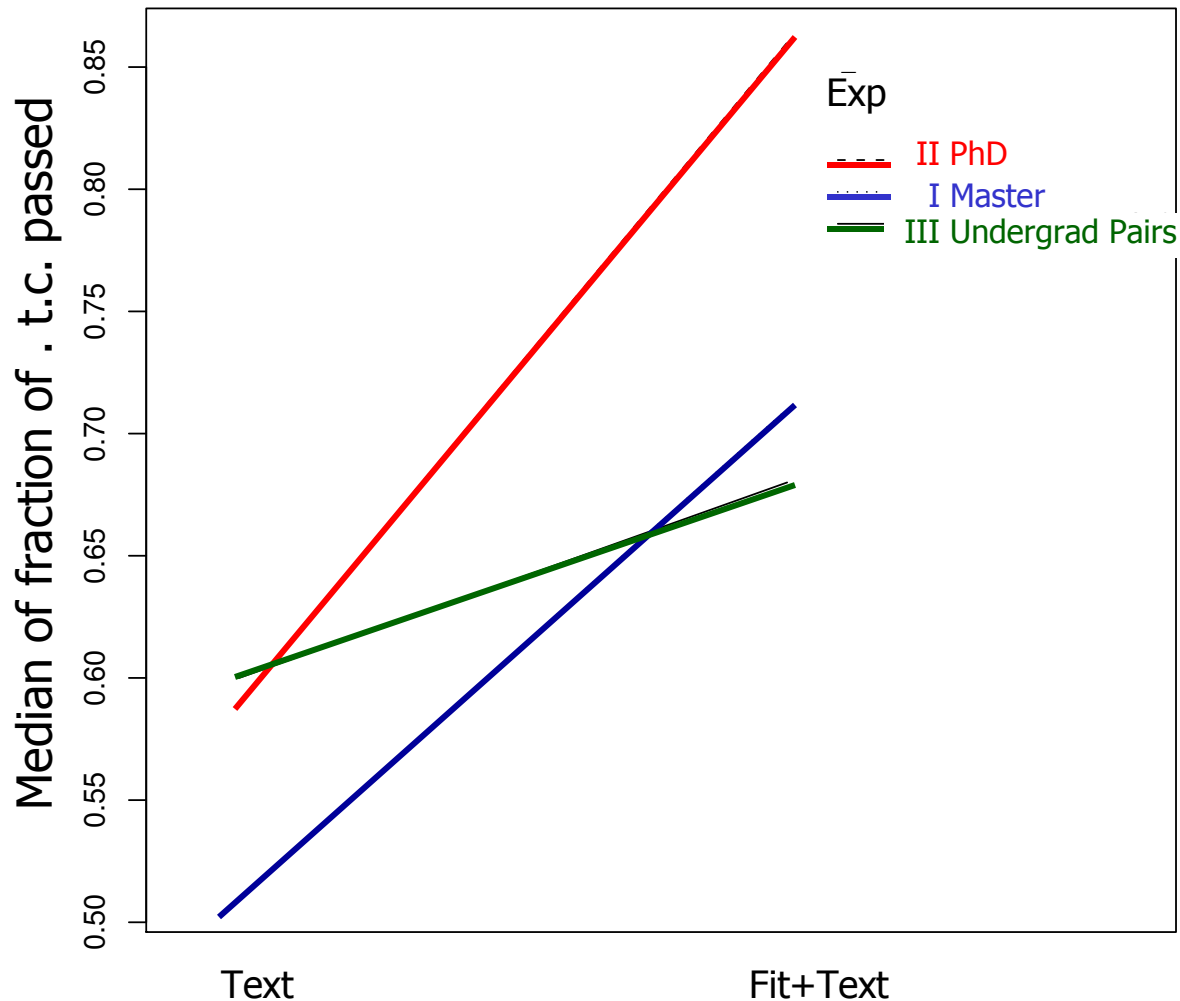
Results: Time



- *Exp I and Exp II:* median and mean times with Fit tables slightly higher than values without Fit tables.
- *Exp III:* subjects with Fit spent less time than subjects without Fit.
- No significant difference found in any experiment.

H_{0t} cannot be therefore rejected

Interaction between Main factor and Experience



- Two-way ANOVA:
 - ◆ the **experience** effect significant (p-value=0.039)
- Subjects with different experience gained different benefits from the use of Fit tables.
- **Slope (benefit) higher for highly experienced subjects**

Threats to validity - I

- **Conclusion validity**
 - ◆ Statistical tests properly used (not violated assumptions)
 - ◆ Small sample size (32 points) → Low statistical power
- **Construct validity**
 - ◆ Code correctness measured in an objective manner
 - ◆ JUnit test suites developed independently from the acceptance test suites.
 - ◆ Time was measured by means of proper time sheets validated by professors/teaching assistants during experiment execution

Threats to validity - II

■ Internal validity

- ◆ Two-way ANOVA was performed to analyze the effect of hypothetical external factors.
- ◆ Learning effect balanced by the design
- ◆ Students were not evaluated on their performance
- ◆ Students participating to each experiment had a similar background
- ◆ **Need a further experiment to separate the effect of undergraduate from the effect of pairs**

■ External validity

- ◆ Simple Java systems chosen
- ◆ Different categories of students (and Universities) but...
- ◆ **Results need to be extended to professionals**

Conclusions of the series of experiments on Fit tables

- **Results indicate:**
 1. higher code correctness with Fit tables
 2. time overhead negligible
 3. Fit produces higher benefits for subjects with a higher experience, such as PhD students
 4. working in pairs seemed to compensate the benefits of Fit in terms of correctness although it may reduce the task time.

References

1. C. Wohlin, P. Runeson, M. Höst, M. C Ohlsson, B. Regnell, A. Wesslén, **Experimentation in Software Engineering - An Introduction**, *Book - Kluwer Academic Press*.
2. F. Ricca, M. Di Penta, M. Torchiano, P. Tonella, M. Ceccato and C. A. Visaggio. **Are Fit tables really talking? a series of experiments to understand whether Fit tables are useful during evolution tasks.** In *Proceedings of the 30th International Conference on Software Engineering (ICSE 2008)*, pages 361-370. IEEE Computer Society, 10-18 May 2008.

Thanks for the attention!



Questions?